pipl | SEARCH API Cheat sheet
REQUEST

Last Update: February 2022

ⓘ See information sheet

Field description, or minimum length
Possible value with **bold** indicating the default value
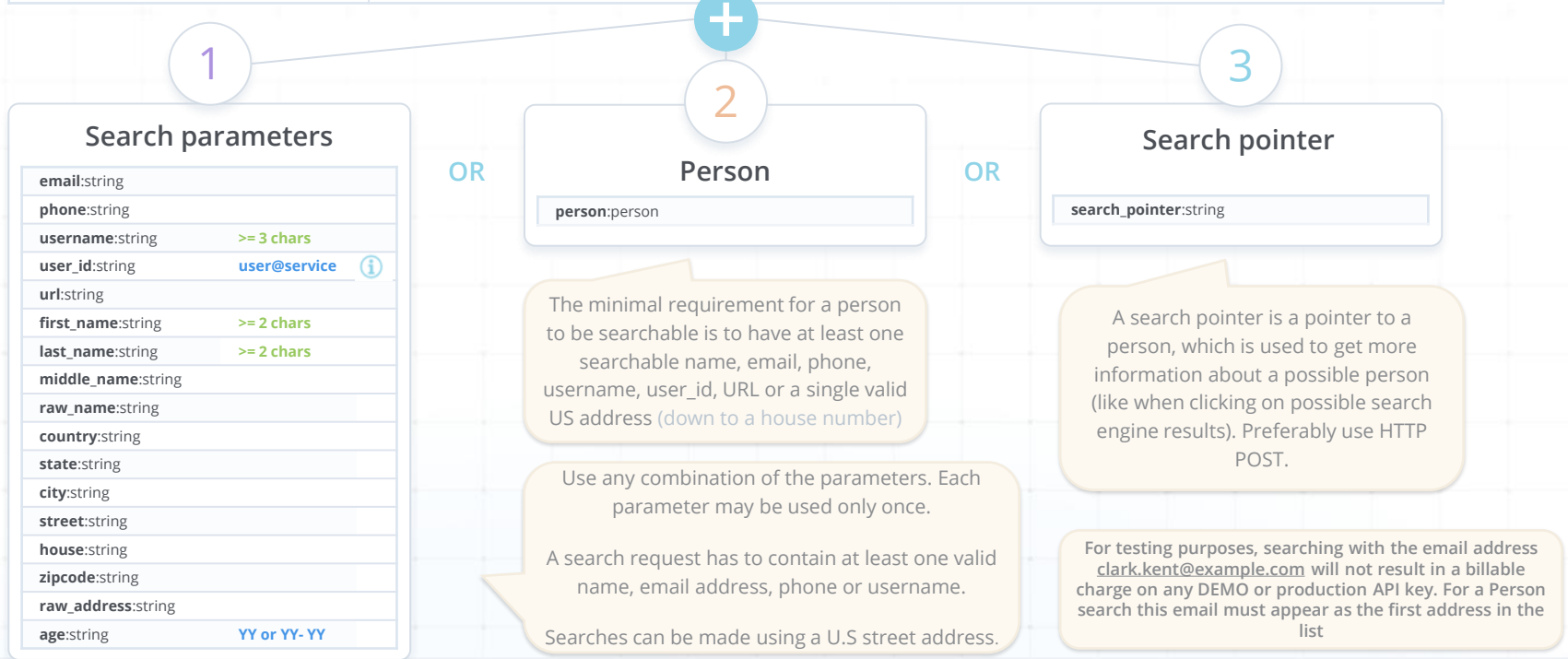
http://api.pipl.com/search/

Accepts a GET request, or POST request with Content-Type: application/x-www-form-urlencoded. All parameters should be UTF-8 and URL Encoded. Supports HTTPS.

SEARCH API REQUEST CONFIGURATION

Field	Description	Values
key:string	Your API key, this is our way to identify and authorize your call	
pretty:bool	The response is "pretty-printed" (with indentation)	**true**/false
minimum_probability:float	The minimum acceptable probability for inferred data	0.0-1.0 **(0.9)**
infer_persons:bool	Persons made up solely from data inferred by statistical analysis from your search query are returned	true/**false**
minimum_match:float	The minimum required match score for possible persons to be returned	**0.0**-1.0
show_sources:string	All: all sources shown; matching/true: only sources from the person; false: don't show sources	true/**false**/all/matching
hide_sponsored:bool	Whether to omit results marked "sponsored" (when additional data from this source is available behind a website paywall)	true/**false**
live_feeds:bool	Whether to use live data sources. Can be turned off for performance	**true**/false
match_requirements:string	A condition to specify what fields you must get back. Response not meeting criteria, return empty and are not charged	
source_category_requirements:string	A condition to specify what source categories are required to exist. Responses not meeting criteria, return empty and are not charged	
callback:string	For JSONP support in JavaScript applications. Only alphanumeric characters and underscores are allowed	

➕

1

Search parameters

email:string	
phone:string	
username:string	>= 3 chars
user_id:string	user@service ⓘ
url:string	
first_name:string	>= 2 chars
last_name:string	>= 2 chars
middle_name:string	
raw_name:string	
country:string	
state:string	
city:string	
street:string	
house:string	
zipcode:string	
raw_address:string	
age:string	YY or YY- YY

OR

2

Person

person:person

The minimal requirement for a person to be searchable is to have at least one searchable name, email, phone, username, user_id, URL or a single valid US address (down to a house number)

Use any combination of the parameters. Each parameter may be used only once.

A search request has to contain at least one valid name, email address, phone or username.

Searches can be made using a U.S street address.

OR

3

Search pointer

search_pointer:string

A search pointer is a pointer to a person, which is used to get more information about a possible person (like when clicking on possible search engine results). Preferably use HTTP POST.

For testing purposes, searching with the email address clark.kent@example.com will not result in a billable charge on any DEMO or production API key. For a Person search this email must appear as the first address in the list

pipl | SEARCH API Cheat sheet
RESPONSE

All fields prefixed with '@' represent metadata

Ⓢ Only shown if show_sources is not false in the request.

★ Possible persons contain Person summary objects, which do not include email addresses or URLs. To get to the full Person object, do an API request with the search pointer of the possible person.
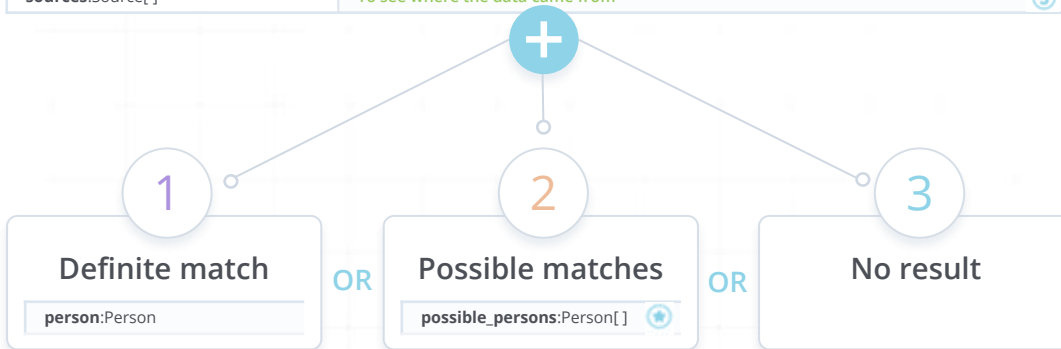
SEARCH API RESPONSE

@https_status_code:int	Status of the response	200/400/403/500
@visible_sources:int	The number of sources returned in the sources array	Ⓢ
@available_sources:int	Number of sources relevant to the search	
@persons_count:int	Number of person objects returned in the response	
@search_id:string	Internal PIPL ID, useful for debugging	
query:Person	Person object representing your search parameters	
match_requirements:string	Canonical expression of requested match_requirements	
available_data:AvailableData[ ]	A summary of data field counters	
error:string	The error message why your search did not run	
warnings:string[ ]	To help you optimize searches and get better results	
sources:Source[ ]	To see where the data came from	Ⓢ

HTTP RESPONSE HEADERS INDICATING QUOTA AND THROTTLING LIMITS

X-APIKey-QPS-Allotted	The number of queries you are allowed to do per second
X-APIKey-QPS-Current	The number of queries you have to run this second
X-QPS-Live-Allotted	The number of live queries you are allowed to do per second.
X-QPS-Live-Current	The number of live queries you have run this second.
X-APIKey-Quota-Allotted	Your quota limit
X-APIKey-Quota-Current	Your used quota
X-Quota-Reset	The time (in UTC) that your quota will be reset

+

Be sure to throttle your queries
By default you can make 10 asynchronous parallel calls per second with live feeds, and 20 asynchronous parallel calls per second without live feeds.
See https://docs.pipl.com/docs/rate-limiting-your-queries

+

1 Definite match
person:Person

OR

2 Possible matches
possible_persons:Person[ ] ★

OR

3 No result

Response Metadata

Available Data
emails:int
relationships:int
usernames:int
user_ids:int
social_profiles:int
jobs:int
addresses:int
phones:int
mobile_phones:int
landline_phones:int
educations:int
names:int
dobs:int
images:int
genders:int
origin_countries:int

A summary indicating the number of data fields that were found

@HTTP_STATUS_CODE	EXPLANATION
200	Successful call
400	Bad Request – either we could not decode your search request, or it did not contain enough data for a search
403	Calls Exceeded – you have either been rate limited or you've reached your quota
500	Any other internal error

Be sure to cater for error handling
We recommend to log all queries to Pipl and review warnings from time to time. A complete list of error messages can be seen at https://docs.pipl.com/reference#errors

pipl

SEARCH API Cheat sheet
CONTAINERS & DATA FIELDS

Legend:
- Request only
- Response only
- Can be used for API search
- Correlate a Person to a Source using id and person_id
- Contact and up
- Social and up
- Business

> The containers are used in both request and responses.
> The data fields have been marked as to what can be searched for.
> Metadata will only be returned in the response.

DataField
- **@valid_since**:Date ⓘ
- **@last_seen**:Date ⓘ
- **@current**:bool ⓘ
- **@inferred**:bool ⓘ

DateRange
- **start**:string   YYYY-MM-DD
- **end**:string   YYYY-MM-DD

Person
- **@id**:GUID ⭐ ⓘ
- **@inferred**:bool ⓘ
- **@search_pointer**:string ⓘ
- **@match**:float ⓘ
- **names**:Name[ ]
- **emails**:Email[ ]
- **usernames**:Username[ ]
- **phones**:Phone[ ]
- **gender**:Gender
- **dob**:Dob
- **languages**:Language[ ]
- **ethinicities**:Ethinicity[ ]
- **origin_countries**:OriginCountry[ ]
- **addresses**:Address[ ]
- **jobs**:Job[ ]
- **educations**:Education[ ]
- **relationship**:Relationship[ ]
- **user_ids**:UserID[ ]
- **images**:Image[ ]
- **urls**:Url[ ]

👤 Name:DataField 🔍
- **@type**:string ⓘ
- **first**:string
- **middle**:string
- **last**:string
- **prefix**:string
- **suffix**:string
- **raw**:string ↗
- **display**:string ↘

⚥ Gender:DataField 🔍
- **content**:string ⓘ

📅 Dob:DataField 🔍
- **date_range**:DateRange
- **display**:string

📍 Address:DataField 🔍
- **@type**:string ⓘ
- **country**:string
- **state**:string
- **city**:string
- **street**:string
- **house**:string
- **apartment**:string
- **zip_code**:string
- **po_box**:string
- **raw**:string ↗
- **display**:string ↘

☎ Phone:DataField 🔍
- **@type**:string ⓘ
- **do_not_call**:bool ↘
- **country_code**:int
- **number**:int
- **extension**:int
- **raw**:string ↗
- **display**:string ↘
- **display_international**:string

💬 Language:DataField
- **language**:string
- **region**:string
- **display**:int

📍 OriginCountry:DataField
- **country**:string

🌐 URL:DataField 🔍
- **@source_id**:string
- **@domain**:string
- **@name**:string
- **@category**:string ⓘ
- **@url**:string
- **@sponsored**:bool

🏷 Tag:DataField
- **date_range**:DateRange
- **display**:string

🌐 Username:DataField 🔍
- **content**:string ⓘ

🌐 UserID:DataField 🔍
- **content**:string ↘ ⓘ

🖼 Image:DataField
- **url**:string
- **thumbnail_token**:string

🔗 Relationship 🔍
- **@type**:string ⓘ
- **@subtype**:string
- **names**:Name[ ]

💼 Job:DataField 🔍
- **title**:string
- **organization**:string
- **industry**:string
- **date_range**:DateRange
- **display**:string ↘

🎓 Education:DataField 🔍
- **degree**:string
- **school**:string
- **date_range**:DateRange
- **display**:string ↘

✉ Email:DataField 🔍
- **@type**:string
- **address**:string
- **address_md5**:string
- **@disposable**:bool ↘
- **@email_provider**:bool ↘

Source
- **@id**:GUID
- **@name**:string
- **@category**:string ⓘ
- **@domain**:string
- **@person_id**:GUID ⭐
- **@match**:float
- **@sponsored**:bool
- **@origin_url**:string
- **@premium**:bool
- **names**:Name[ ]
- **emails**:Email[ ]
- **usernames**:Username[ ]
- **phones**:Phone[ ]
- **gender**:Gender
- **dob**:Dob
- **languages**:Language[ ]
- **ethnicities**:Ethnicity[ ]
- **origin_countries**:OriginCountry[ ]
- **addresses**:Address[ ]
- **jobs**:Job[ ]
- **educations**:Education[ ]
- **relationship**:Relationship[ ]
- **user_ids**:UserId[ ]
- **images**:Image[ ]
- **urls**:Url[ ]
- **tags**:Tag[ ]

Tag:DataField
- **@classification**:string
- **content**:string

pipl

SEARCH API Cheat sheet
INFORMATION SHEET

> Atomic expressions (field name or source category) can be combined using operators '&' 'and' '|' 'OR', including brackets '(' ')' to separate logic

SEARCH API REQUEST

PARAMETER	EXPLANATION
match_requirements e.g.: (names and addresses) or emails	You can use any one (or more) of these field names in your condition as the most atomic expression: name, address, phone, email, job, education, image, username, user_id, dob, url, gender, ethnicity, language, origin_country, social_profile Plurals can also be used (e.g. names, addresses) The criteria will be met if ANY person in the response contains the requested field. 1. You can append ".TYPENAME" to any field name, where TYPENAME is any recognized @type value of the data field. For example phone.mobile or address.work. Please see the Data Fields table for a summary of types 2. You can prefix any field name with "new_", in which case the criteria will only be met if the response includes data of that type which was not included in the original query.
source_category_requirements e.g.: web_pages or media	You can use any of the source categories as the atomic expression. The criteria will be met if the response contains a source of the requested category (please note that in order to see the source, you need to set the show_sources parameter).

DATA SOURCE CATEGORY

CATEGORY	EXAMPLES
personal_profiles	facebook.com, pinterest.com, last.fm.
media	picasa.com, flickr.com, youtube.com
professional_and_business	linkedin.com, spoke.com, zoominfo.com
public_records	bop.gov, archives.com, ancestry.com
publications	uspto.gov, portal.acm.org, scirus.com
school_and_classmates	classmates.com, reunion.com, schoolbank.nl
email_address	intelius.com, spokeo.com
background_reports	peoplefinders.com intelius.com peoplesmart.com
contact_details	whitepages.com, addresses.com, anywho.com
web_pages	Default category, for sources that don't match any of the above categories.

SEARCH PARAMETERS

PARAMETER	DESCRIPTION
user_id Format is userid@service	Supported services include: facebook, linkedin, twitter, myspace, foursquare, ebay, google, pinterest, instagram, lastfm, flickr, youtube, digg tagged, hi5, weeworld, delicious, hyves, deviantart, douban, odnoklassniki, quora, xanga, mylife, gaia, stumbleupon, bebo, livejournal, viadeo, ning, myheritage, qzone, vkontakte, flixster, tumblr, myyearbook, badoo, habbo, xing, sonico, friendster, meetup, goodreads, classmates, renren, cyworld, netlog, soundcloud, yelp, orkut, aboutme, flavorsme, freelancer, gravatar, imgur, github, CPF, medium, NPI, tiktok, strava, academia, VIN.

DATA FIELDS

DATA FIELD	COMPONENT	POSSIBLE VALUES (DEFAULT MARKED IN BOLD)
Name	@type	**present**/maiden/former
Address	@type	**home**/work/old
Phone	@type	mobile/home_phone/home_fax/work_phone/pager
Gender	content	male/female
URL	@category	'background_reports', 'contact_details', 'email_address', 'media', 'personal_profiles', 'professional_and_business', 'public_records', 'publications', 'school_and_classmates', 'web_pages'
Relationship	@type	work/family/**friend**/other
UserID	content	Format: user_id@service See list of supported services above
Email	@type	**personal**/work
Tag	@classificatiion	skill, course, volunteering - (professional credentials) following, followers - (social media follower/following count) auto year, auto make, auto model, vin - (vehicle information) associated domain - (website ownership information) death date, burial address - (death information)

For more and detailed information about the Pipl SEARCH API please refer to the reference guide at https://docs.pipl.com/reference

pipl

SEARCH API Cheat sheet
THUMBNAIL API

☑ Cropping images to the right dimensions

☑ Zooming in on faces with advanced face detection

☑ Attributing sources by adding favicons to images

☑ Image redundancy - specify a fallback image

http://thumb.pipl.com/image?

THUMBNAIL API REQUEST URL PARAMETERS

tokens:string	One or two thumbnail tokens, separated by a comma. If the image referenced by the first thumbnail token is not available on the web any more, the second token will be used	FIRST_TOKEN{,SECOND_TOKEN}
height:int	Height in pixels	100-500
width:int	Width in pixels	100-500
favicon:bool	Whether to show favicon	**true**/false
zoom_face:bool	Whether to enable face zoom	true/**false**

🖼 **Image**:DataField
url:string
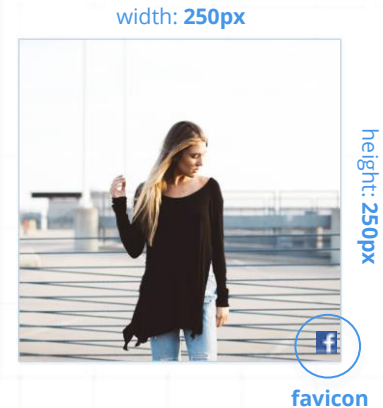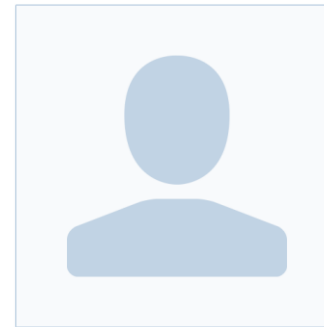**thumbnail_token**:string

Thumbnail API example

http://thumb.pipl.com/image?height=250&width=250&favicon=true&zoom_face=true&tokens=AE2861B242686E7BD0CB4D9049298EB7D18FEF66D950E8AB78BCD3F484345CE74536C19A85D0BA3D32DC9E7D1878CD4D341254E7AD129255C6983E6E154C4530A0DAAF665EA325FC0206F8B1D7E0B6B7AD9EBF71FCF610D57D

A redundant thumbnail URL example

http://thumb.pipl.com/image?height=250&width=250&favicon=true&zoom_face=true&tokens=FIRST_TOKEN,SECOND_TOKEN

Sometimes, images will no longer be available on the web. To help you create robust thumbnail URLs, our thumbnail API allows specifying a fallback token.

Special cases

width: **250px**

height: **250px**

favicon

In some cases, both of your tokens will reference an image no longer available online - or they may contain a default placeholder image, and not a real image of the person.

In those cases, the thumbnail server will return 404.

For more and detailed information about the SEARCH thumbnail API, please refer to https://docs.pipl.com/docs/thumbnails